Xbox Software Hacking

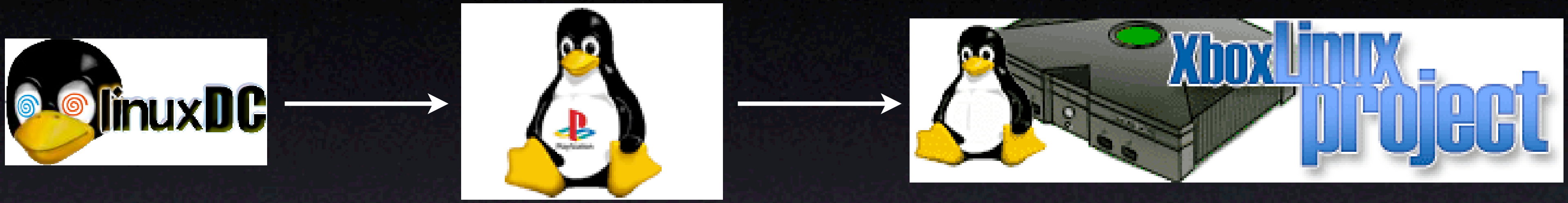
Michael Steil, David Jilli, Stefan Esser,
Franz Lehner, Jeff Mears, Edgar Hucek

http://www.xbox-linux.org/

Agenda

- Xbox Linux status Dec 2002
- the 007 hack (”Agent Under Fire”)
- the Ernie & Bert hack (Dashboard)
- the Audio hack (Dashboard)
- MechInstaller
- demonstration, future

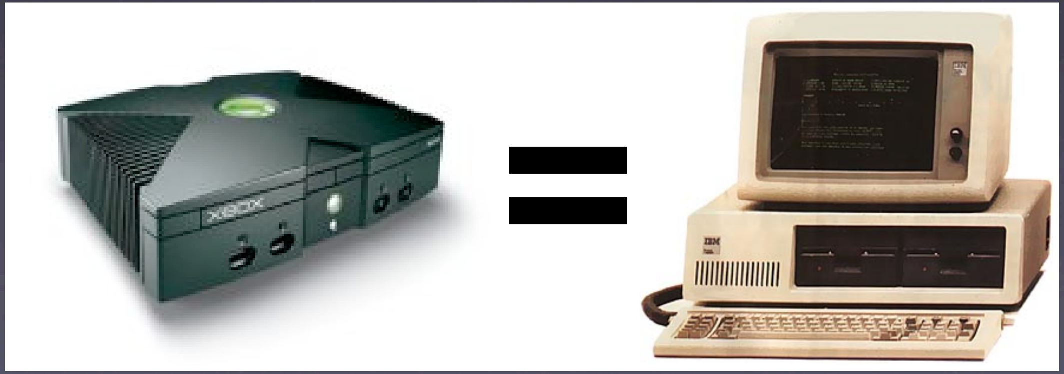
Xbox Linux Motivation



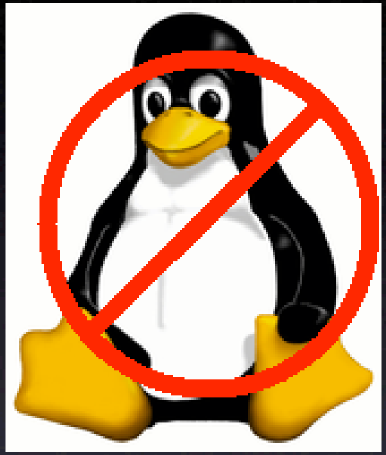
- Linux for Dreamcast
- Linux for Playstation 2
- “*The Xbox is a great gaming console*”!?

The Xbox is a PC

- Celeron III Coppermine 733
- 64 MB RAM
- nVidia GeForce 3MX
- 10 GB IDE hard disk, IDE DVD
- 10/100 Ethernet
- 4x USB 1.1

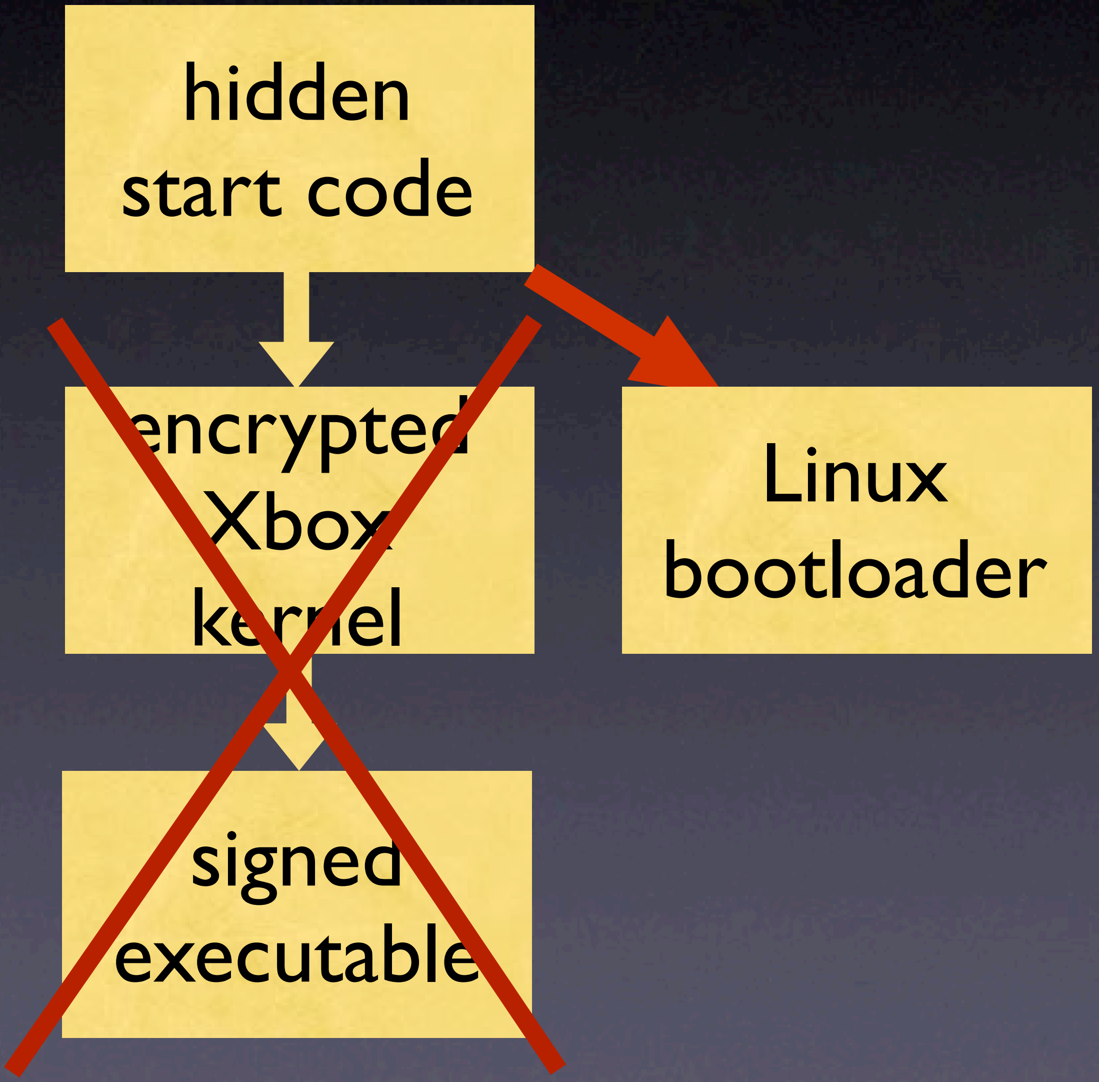


Anti-Linux-Protection?

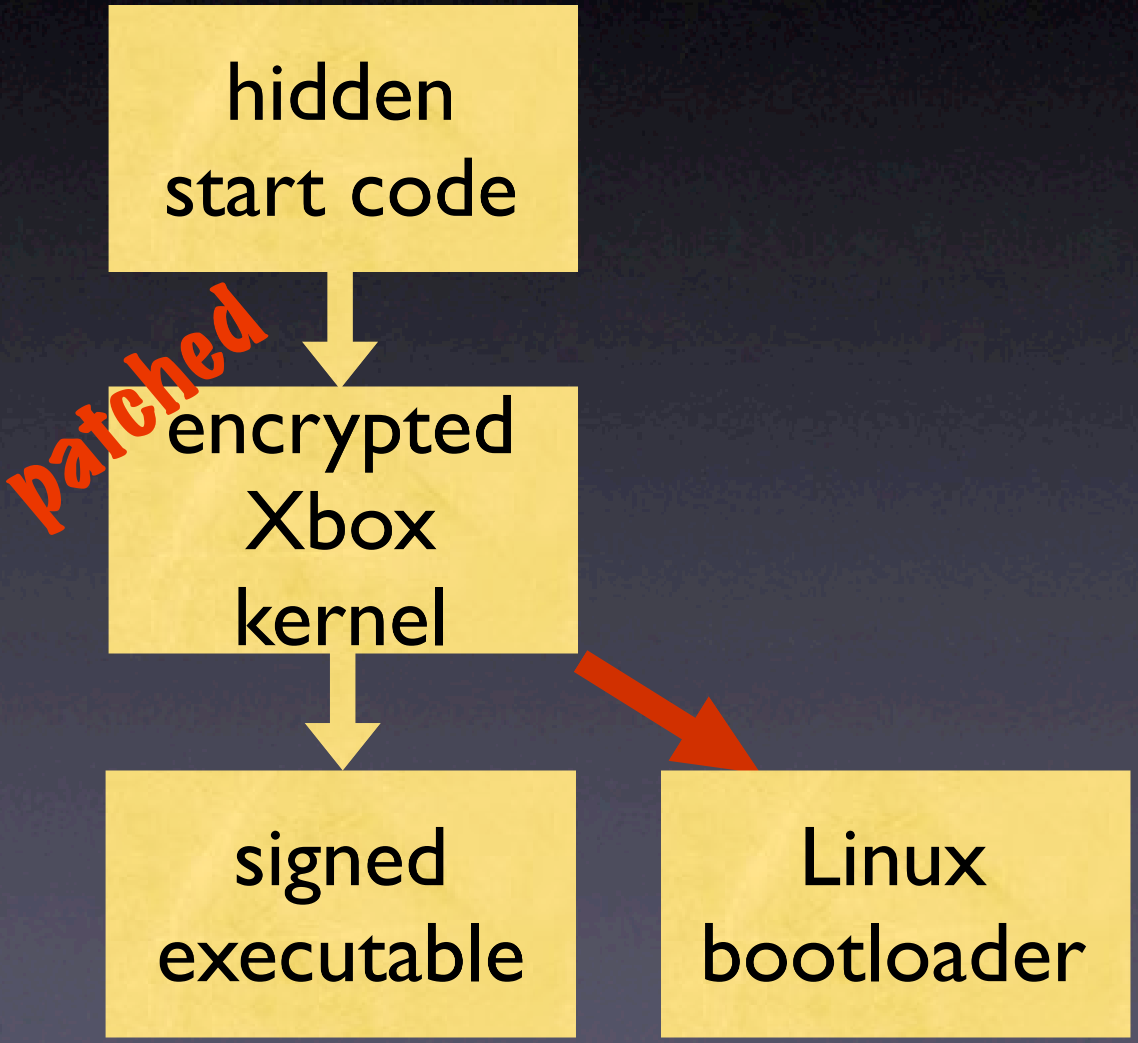


- sophisticated security system
- custom DVD format
- accepts only signed executables (XBE)
- chain of trust
- homebrew-in-sandbox solution would have been possible

Chain of Trust



Chain of Trust



Xbox Linux History

- **23 May 2002**: founded
- **13 Aug 2002**: kernel booted
- **07 Oct 2002**: distro with KDE & Gnome
- **17 Dec 2002**: Linux bootloader ROM

Modchips

- Xbox had to be opened
- warranty problem
- modchips were > 30 EUR/USD
- too complicated for many people
- modchips are associated with piracy
- **software-only solution needed!**

Approaches

- Break the RSA-2048?
- Look for vulnerabilities in the software?

007™

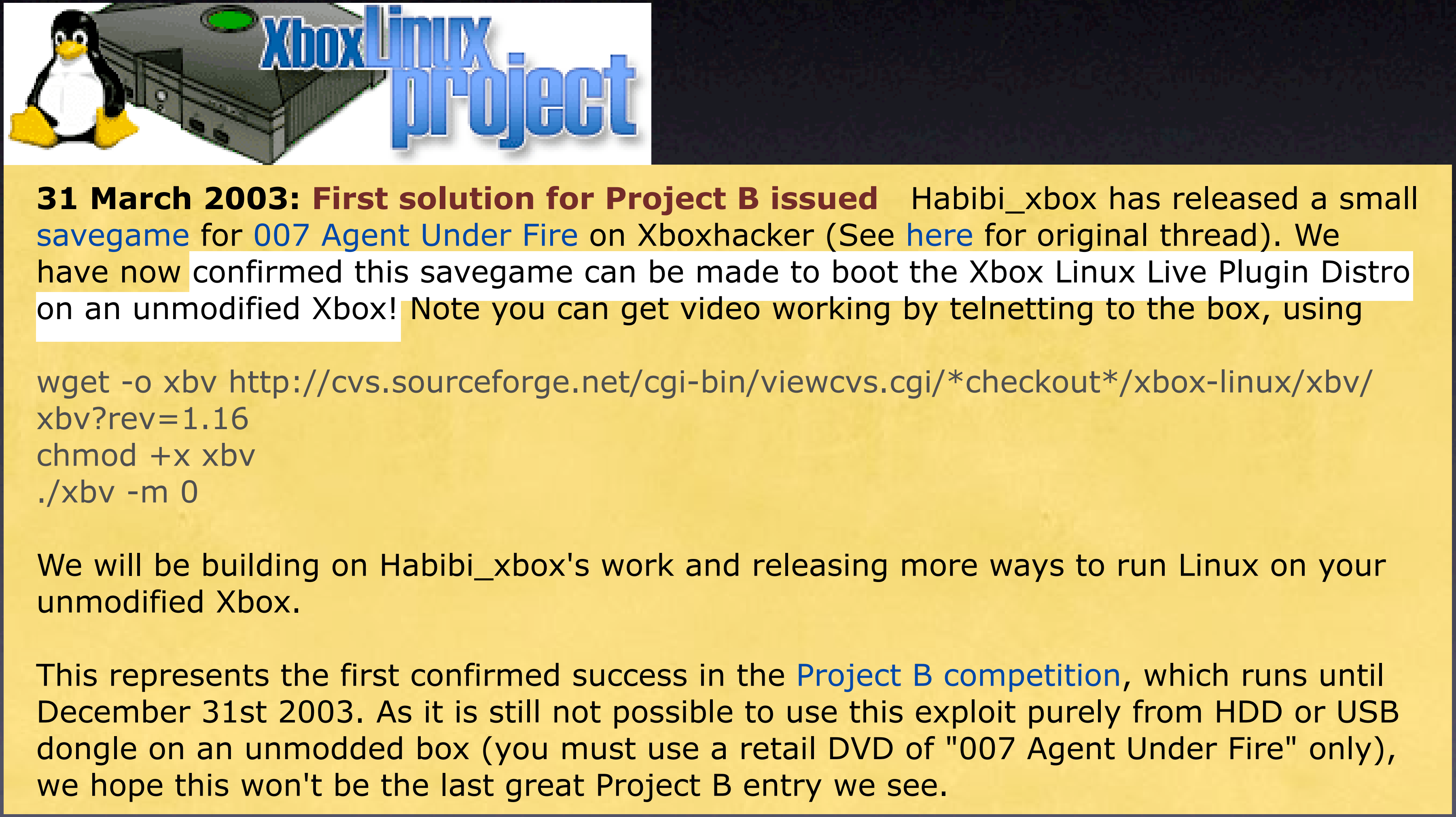
The 007 hack

From: habibi_xbox
Date: 30 Mar 2003
Subject : Project B Solved !

Ladies and Gentlemen,
I'm happy to present the first solution found for the Xbox Linux Project B:
Here is a way to run Xbox Linux on an unmodded, unopened Xbox !
Inlcuded is a uuencoded zip file containing all the necessary files. Here is what you need:
- - You need an unmodded XBOX (not sure it works with modded bios)
- - You need the game 007 Agent Under Fire (*NOT* NIGHTFIRE, those are two different games!)
- - You need a way to transfer a save to a memory card (that is, xbox-save.com's hardware, or usb<>xbox cable + usb stick + xbox-save software, or you can use a standard memory card too if you can put files on it (with EvoX for instance).
- - You need to get the "Xbox Linux Live" small distro.

Got all this? Let's party!

The 007 hack



31 March 2003: **First solution for Project B issued** Habibi_xbox has released a small savegame for 007 Agent Under Fire on Xboxhacker (See here for original thread). We have now confirmed this savegame can be made to boot the Xbox Linux Live Plugin Distro on an unmodified Xbox! Note you can get video working by telnetting to the box, using

```
wget -o xbv http://cvs.sourceforge.net/cgi-bin/viewcvs.cgi/*checkout*/xbox-linux/xbv/
xbv?rev=1.16
chmod +x xbv
./xbv -m 0
```

We will be building on Habibi_xbox's work and releasing more ways to run Linux on your unmodified Xbox.

This represents the first confirmed success in the Project B competition, which runs until December 31st 2003. As it is still not possible to use this exploit purely from HDD or USB dongle on an unmodded box (you must use a retail DVD of "007 Agent Under Fire" only), we hope this won't be the last great Project B entry we see.

David Jilli

- Swiss Federal Institute of Technology student
- Interests in computer security, low level programming and video game systems
- Xbox-Linux contribution: 007 save hack.

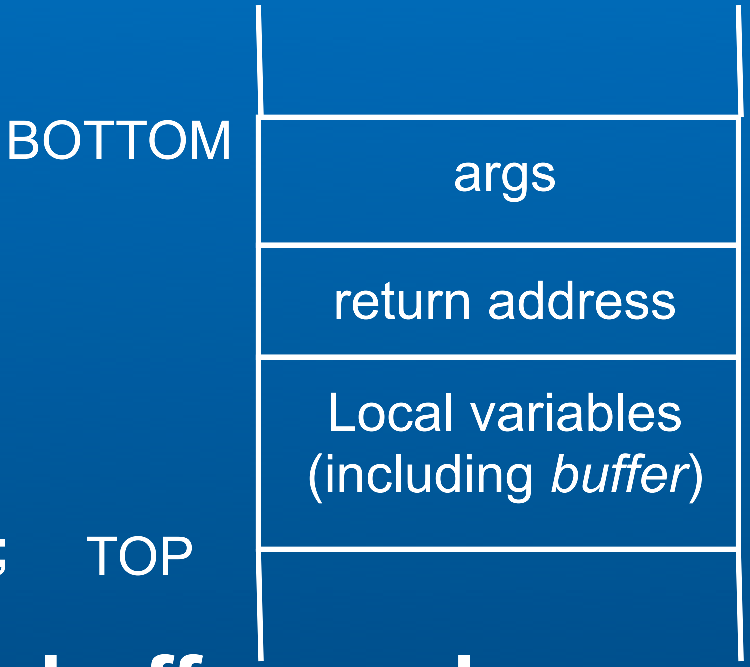
About buffer overflows (1)

- Way to execute code not intended to run
- Widely used to gain root access using vulnerable programs
- Stuffing more data into a buffer than it can handle
- Usually triggered by badly formed input

About buffer overflows (2)

- Usually happens on the stack

```
void function(char* name)
{
  char buffer[20];
  sprintf(buffer, “Hello %s”, name);
  …
}

function (”this name is too biiiiiiiiiig”);
```



- Data can go out of the buffer, and return address can be changed

Buffer overflows for Xbox Linux

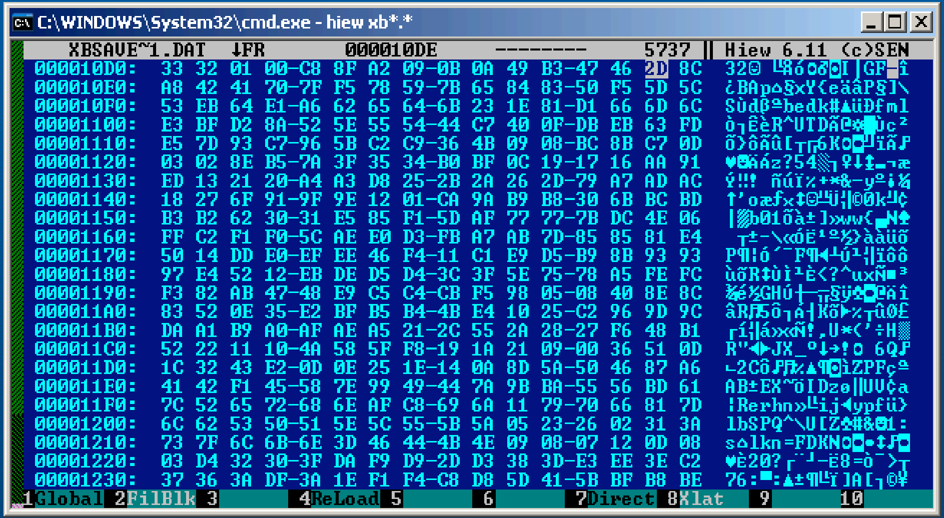
- If a buffer overflow can be triggered on the Xbox, code can run without a modchip
- What « badly formed input » can we provide ?
- Idea : find game saves with strings in them
- sprintf() overflows can be triggered if input not properly checked

Xbox Game Saves

- Size : from a few bytes to several Megabytes
- ~ 2/3 of them contain strings
- Problem : can’t be edited (hashed)
- Hashing done by the game itself so
  – Hash check can be removed from the game (MS library function) for development
  – Correct hash is computed by the game for comparison (so can be dumped)

What's needed then…

- Understanding of the Xbox architecture
- A way to dump memory
- A way to catch exceptions
- Games
- And some luck…



C:\WINDOWS\System32\cmd.exe - hiew xb*.*

```
   XBSAVE~1.DAT  ↓FR        000010DE    ---------       5737 ‖ Hiew 6.11 (c)SEN
000010D0:  33 32 01 00-C8 8F A2 09-0B 0A 49 B3-47 46 2D 8C   32☺ ╚Åó○♂◙I│GF-î
000010E0:  A8 42 41 70-7F F5 78 59-7B 65 84 83-50 F5 5D 5C   ¿BAp⌂§xY{eäâP§]\
000010F0:  53 EB 64 E1-A6 62 65 64-6B 23 1E 81-D1 66 6D 6C   Sùdß ªbedk#▲üDfml
00001100:  E3 BF D2 8A-52 5E 55 54-44 C7 40 0F-DB EB 63 FD   ò┐ÊèR^UTDÃ@☼█Ùc²
00001110:  E5 7D 93 C7-96 5B C2 C9-36 4B 09 08-BC 8B C7 0D   õ}ôÃû[┬╔6K○◘╝ïÃ♪
00001120:  03 02 8E B5-7A 3F 35 34-B0 BF 0C 19-17 16 AA 91   ♥☻ÄÁz?54░┐♀↓↨▬¬æ
00001130:  ED 13 21 20-A4 A3 D8 25-2B 2A 26 2D-79 A7 AD AC   ý‼! ñúÏ%+*&-yº¡¼
00001140:  18 27 6F 91-9F 9E 12 01-CA 9A B9 B8-30 6B BC BD   ↑'oæƒ×↕☺╩Ü╣©0k╝¢
00001150:  B3 B2 62 30-31 E5 85 F1-5D AF 77 77-7B DC 4E 06   │▓b01õà±]»ww{▄N♠
00001160:  FF C2 F1 F0-5C AE E0 D3-FB A7 AB 7D-85 85 81 E4    ┬±-\«óË¹º½}ààüõ
00001170:  50 14 DD E0-EF EE 46 F4-11 C1 E9 D5-B9 8B 93 93   P¶¦ó´¯F¶◄┴Ú ı╣ïôô
00001180:  97 E4 52 12-EB DE D5 D4-3C 3F 5E 75-78 A5 FE FC   ùõR↕ÙÌıÈ<?^uxÑ■³
00001190:  F3 82 AB 47-48 E9 C5 C4-CB F5 98 05-08 40 8E 8C   ¾é½GHÚ┼─╦§ÿ♣◘@Äî
000011A0:  83 52 0E 35-E2 BF B5 B4-4B E4 10 25-C2 96 9D 9C   âR♫5Ô┐Á┤Kõ►%┬û Ø£
000011B0:  DA A1 B9 A0-AF AE A5 21-2C 55 2A 28-27 F6 48 B1   ┌í╣á»«Ñ!,U*('÷H▒
000011C0:  52 22 11 10-4A 58 5F F8-19 1A 21 09-00 36 51 0D   R"◄►JX_°↓→!○ 6Q♪
000011D0:  1C 32 43 E2-0D 0E 25 1E-14 0A 8D 5A-50 46 87 A6   ∟2CÔ♪♫%▲¶◙ìZPFçª
000011E0:  41 42 F1 45-58 7E 99 49-44 7A 9B BA-55 56 BD 61   AB±EX~ÖIDzø║UV¢a
000011F0:  7C 52 65 72-68 6E AF C8-69 6A 11 79-70 66 81 7D   |Rerhn»╚ij◄ypfü}
00001200:  6C 62 53 50-51 5E 5C 55-5B 5A 05 23-26 02 31 3A   lbSPQ^\U[Z♣#&☻1:
00001210:  73 7F 6C 6B-6E 3D 46 44-4B 4E 09 08-07 12 0D 08   s⌂lkn=FDKN○◘•↕♪◘
00001220:  03 D4 32 30-3F DA F9 D9-2D D3 38 3D-E3 EE 3E C2   ♥È20?┌¨┘-Ë8=Ò¯>┬
00001230:  37 36 3A DF-3A 1E F1 F4-C8 D8 5D 41-5B BF B8 BE   76:▀:▲±¶╚Ï]A[┐©¥
1Global 2FilBlk 3       4ReLoad 5       6       7Direct 8Xlat   9       10
```

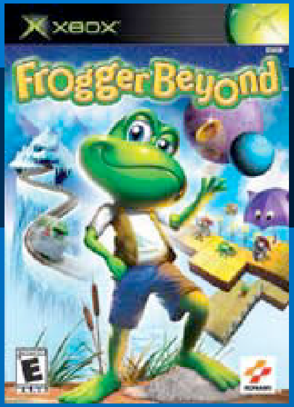
Let’s play…

- Started replacing strings with much larger ones
- ~1/3 of games checked at first are crashing
- Xbox crash : nothing special happens
- crashing ≠ exploitable

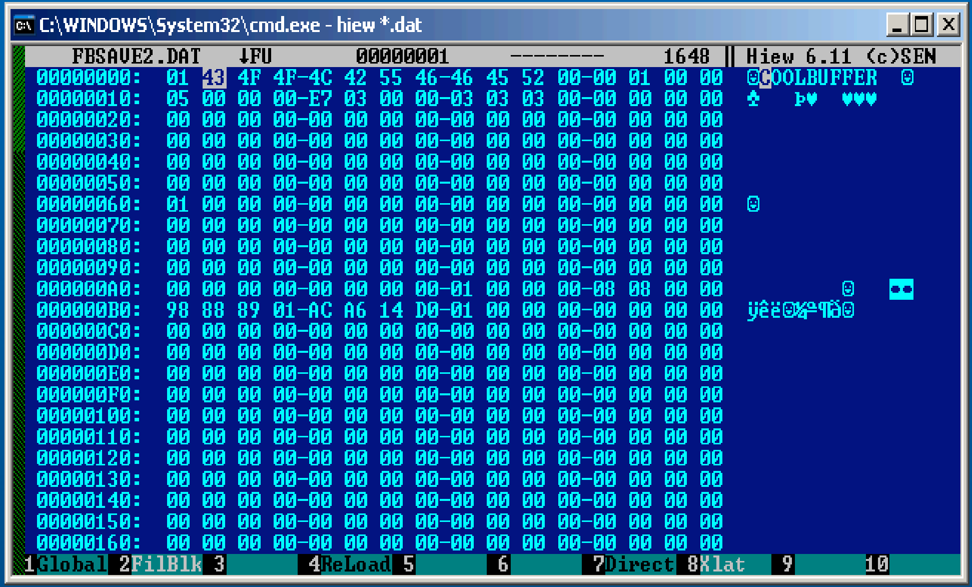
The « Amped » case

- One of the first game checked
- Lots of strings
- Easy to trace exceptions (caught by the game, error display)
- Return address can't be changed *but* some processor registers can be set to any value (but aren't used…)
- Means the save game idea was good

The « Frogger » case (1)

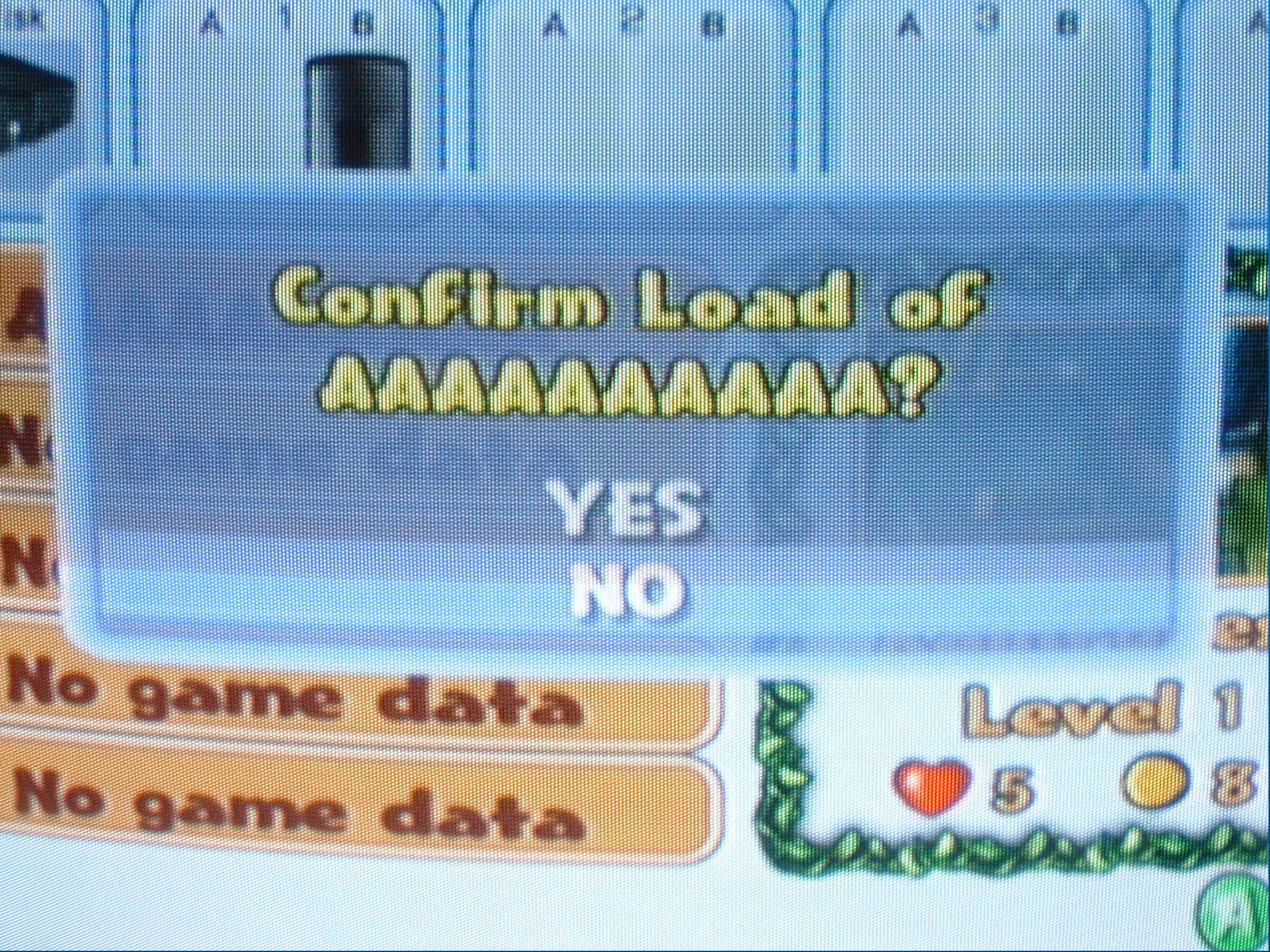


- Just one string
- Player name: 10 chars
- sprintf(buffer, “Confirm load of %s?”, player_name);
- Can be overflowed but return address can’t be changed directly



C:\WINDOWS\System32\cmd.exe - hiew *.dat

```
    FBSAVE2.DAT    ↓FU          00000001      ---------      1648 || Hiew 6.11 (c)SEN
 00000000:  01 43 4F 4F-4C 42 55 46-46 45 52 00-00 01 00 00  ☺COOLBUFFER  ☺
 00000010:  05 00 00 00-E7 03 00 00-03 03 03 00-00 00 00 00  ♣   þ♥  ♥♥♥
 00000020:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
 00000030:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
 00000040:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
 00000050:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
 00000060:  01 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00  ☺
 00000070:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
 00000080:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
 00000090:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
 000000A0:  00 00 00 00-00 00 00 00-01 00 00 00-08 08 00 00          ☺   ◘◘
 000000B0:  98 88 89 01-AC A6 14 D0-01 00 00 00-00 00 00 00  ÿêë☺¼ª¶Ð☺
 000000C0:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
 000000D0:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
 000000E0:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
 000000F0:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
 00000100:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
 00000110:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
 00000120:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
 00000130:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
 00000140:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
 00000150:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
 00000160:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
1Global 2FilBlk 3       4ReLoad 5      6       7Direct 8Xlat   9       10
```



Confirm Load of AAAAAAAAAAA?

YES

NO

No game data

No game data

Level 1

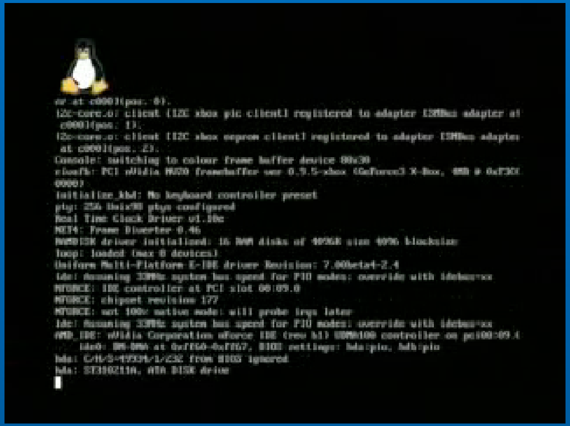
♥ 5 ● 8

The « Frogger » case (2)

- With some hacking, it’s enough to jump to the place we want!
  - jmp [eax+10h] where we can control eax
- Shellcode is put into the save
- Save is always loaded at the same address, can jump to a constant address
- Shellcode applies modchip-like BIOS patches and runs a program

The « Frogger » case (3)

- Hello Tux !



```
er at c000](pos: 0).
i2c-core.o: client [I2C xbox pic client] registered to adapter [SMBus adapter at
 c000](pos: 1).
i2c-core.o: client [I2C xbox eeprom client] registered to adapter [SMBus adapter
 at c000](pos: 2).
Console: switching to colour frame buffer device 80x30
rivafb: PCI nVidia NV20 framebuffer ver 0.9.5-xbox (GeForce3 X-Box, 4MB @ 0xF3C0
0000)
initialize_kbd: No keyboard controller preset
pty: 256 Unix98 ptys configured
Real Time Clock Driver v1.10e
NET4: Frame Diverter 0.46
RAMDISK driver initialized: 16 RAM disks of 4096K size 4096 blocksize
loop: loaded (max 8 devices)
Uniform Multi-Platform E-IDE driver Revision: 7.00beta4-2.4
ide: Assuming 33MHz system bus speed for PIO modes; override with idebus=xx
NFORCE: IDE controller at PCI slot 00:09.0
NFORCE: chipset revision 177
NFORCE: not 100% native mode: will probe irqs later
ide: Assuming 33MHz system bus speed for PIO modes; override with idebus=xx
AMD_IDE: nVidia Corporation nForce IDE (rev b1) UDMA100 controller on pci00:09.0
    ide0: BM-DMA at 0xff60-0xff67, BIOS settings: hda:pio, hdb:pio
hda: C/H/S=49334/1/232 from BIOS ignored
hda: ST310211A, ATA DISK drive
█
```

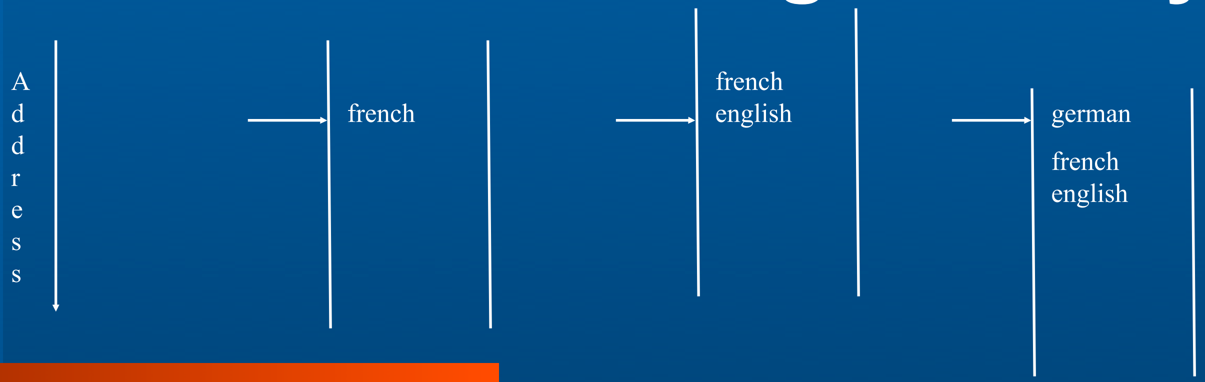
- First successful boot of Xbox-Linux on an unmodded Xbox
- Problem: Frogger is not released in Europe…
- Hacked save is BIOS specific

The « 007 » case (1)

- Basic overflow on a string buffer
- A save game name > 256 bytes crashes the Xbox
- return address right after those 256 bytes
- Better than Frogger: Game exists in PAL and NTSC, and exploit is not BIOS specific

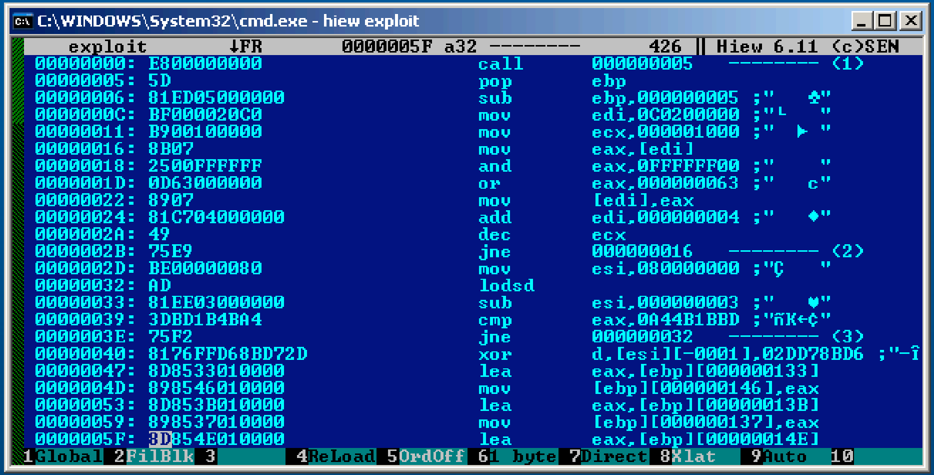
The « 007 » case (2)

- Problem: 4 different executables for 4 different languages
- For each executable, the save is loaded at some slightly different place in RAM
- But one save is enough: 4 entry points



The « 007 » case (3)

- The *shellcode*
  - Turns off kernel write protection
  - Replaces Xbox public key
  - Executes a program signed with a custom private key (Xbox-Linux loader)

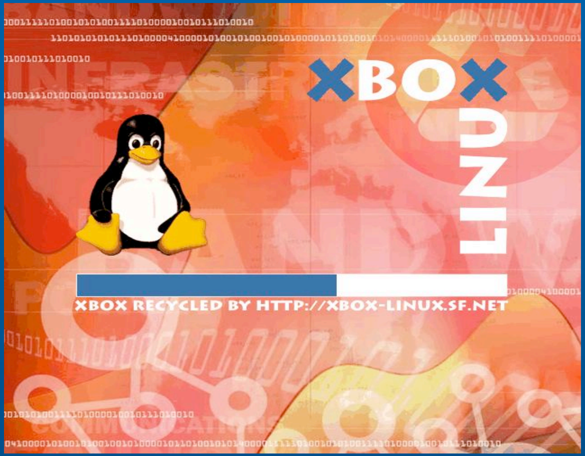


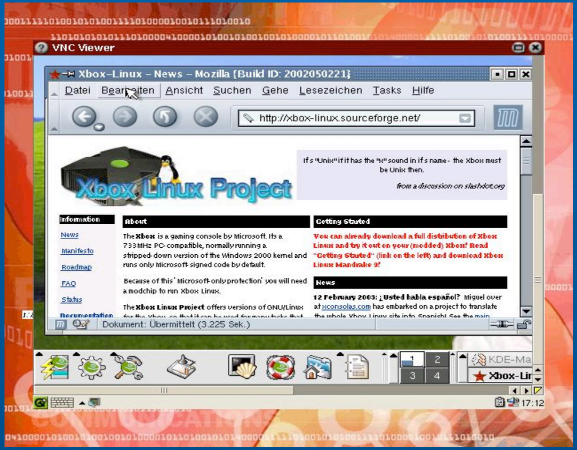
C:\WINDOWS\System32\cmd.exe - hiew exploit

```
   exploit          ↓FR         0000005F a32 ---------         426 ‖ Hiew 6.11 (c)SEN
 00000000: E800000000                       call    000000005    --------- (1)
 00000005: 5D                               pop     ebp
 00000006: 81ED05000000                     sub     ebp,000000005 ;"    ♣"
 0000000C: BF000020C0                       mov     edi,0C0200000 ;"└    "
 00000011: B900100000                       mov     ecx,000001000 ;"   ► "
 00000016: 8B07                             mov     eax,[edi]
 00000018: 2500FFFFFF                       and     eax,0FFFFFF00 ;"     "
 0000001D: 0D63000000                       or      eax,000000063 ;"    c"
 00000022: 8907                             mov     [edi],eax
 00000024: 81C704000000                     add     edi,000000004 ;"    ♦"
 0000002A: 49                               dec     ecx
 0000002B: 75E9                             jne     000000016    --------- (2)
 0000002D: BE00000080                       mov     esi,080000000 ;"Ç    "
 00000032: AD                               lodsd
 00000033: 81EE03000000                     sub     esi,000000003 ;"    ♥"
 00000039: 3DBD1B4BA4                       cmp     eax,0A44B1BBD ;"ñK←Ç"
 0000003E: 75F2                             jne     000000032    --------- (3)
 00000040: 8176FFD68BD72D                   xor     d,[esi][-0001],02DD78BD6 ;"–Î
 00000047: 8D8533010000                     lea     eax,[ebp][000000133]
 0000004D: 898546010000                     mov     [ebp][000000146],eax
 00000053: 8D853B010000                     lea     eax,[ebp][00000013B]
 00000059: 898537010000                     mov     [ebp][000000137],eax
 0000005F: 8D854E010000                     lea     eax,[ebp][00000014E]
1Global 2FilBlk 3        4ReLoad 5OrdOff 61 byte 7Direct 8Xlat   9Auto   10
```

The « 007 » case (4)

- Ensure only Xbox-Linux can be run using this hack
- Xbox-Linux Live was signed




VNC Viewer
Xbox-Linux – News – Mozilla {Build ID: 2002050221}
Datei Bearbeiten Ansicht Suchen Gehe Lesezeichen Tasks Hilfe
http://xbox-linux.sourceforge.net/
Xbox Linux Project
Information
About
Getting Started
News
Manifesto
Roadmap
FAQ
Status
Dokument: Übermittelt (3.225 Sek.)
KDE-Ma
Xbox-Lir
17:12

- Obfuscation

007 Shell Code

```
    ; replace MS RSA key with our own one
    mov esi, 80000000h
se: lodsd
    sub esi,3
    cmp eax,0a44b1bbdh
    jnz se
    xor dword [esi-1],02dd78bd6h
```

```
; map D: to E:
lea eax,[ebp+filename2]
push eax
mov ebx,[ds:ebp+IoDeleteSymbolicLink]
call [ebx]
lea eax,[ebp+filename3]
push eax
lea eax,[ebp+filename2]
push eax
mov ebx,[ds:ebp+IoCreateSymbolicLink]
call [ebx]
```

```
push 0h
lea eax,[ebp+xbename]
push eax
call [ebp+xbelaunch]        ; call the xbe!
```

007 Obfuscation

```
;-------------- evox 2.5 patching --------------
    cld
    mov       edi,0800259AFh
    mov       eax,0x42b69f5e
    stosd
    mov       eax,0x2579952c
    stosd
    mov       al,0A4h
    stosb
    mov       edi,0800259B3h
    mov       eax,0x6937bada
    stosd
    [...]
;-------------- end evox 2.5 patching --------------
    push 0h
    lea eax,[ebp+xbename]
    push eax
    call 0x2a4c6        ; call the xbe!
xbename        db "d:\UDATA\4541000d\000000000000\linux.xbe",0
```

```
0000:1080 ff d8 ff e0 00 10 4a 46 49 46 00 01 01 01 00 47 ˇÿˇ‡..JFIF.....G
0000:1090 00 47 00 00 ff fe 01 02 ad 53 fe 7d 78 85 0d 16 .G..ˇ˛..≠S˛}x...
0000:10a0 f8 05 c9 95 5d f5 4d 19 c0 84 3c a4 ab c0 d7 2f ¯.….]ıM.¿.<§´¿◊/
0000:10b0 2a f7 fa fa 6a 45 3c f2 b6 e7 c1 8e 83 10 4b f3 *˜˙˙jE<Ú∂Á¡...KÛ
0000:10c0 32 ca a3 19 e2 ef 5b cb f6 d8 88 5e a1 8c 6b c0 2 £.,Ô[Àˆÿ.^°.k¿
0000:10d0 eb 8e 40 c3 d9 85 fb 41 e8 b2 fc ff ff c0 67 17 Î.@√Ÿ.˚AË≤¸ˇˇ¿g.
0000:10e0 bb 4d 4c 4b fa 69 fc be e9 ba 2c bd 42 41 40 aa ªMLK˙i¸æÈ∫,ΩBA@™
```

007 Obfuscation

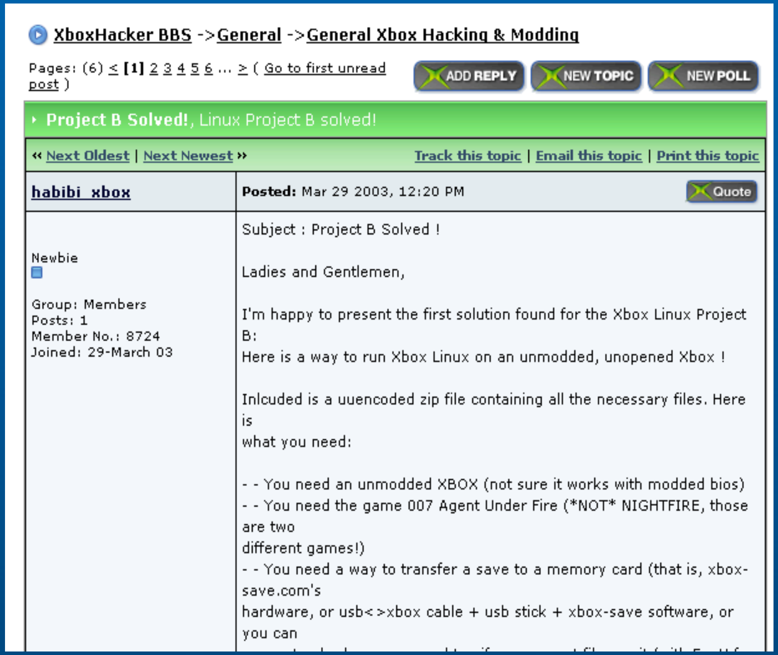
```
;-------------- evox 2.5 patching --------------
    cld
    mov         edi,0800259AFh
    mov         eax,0x42b69f5e
    stosd
    mov         eax,0x2579952c
    stosd
    mov         al,0A4h
    stosb
    mov         edi,0800259B3h
    mov         eax,0x6937bada
    stosd
    [...]
;-------------- end evox 2.5 patching --------------
    push 0h
    lea eax,[ebp+xbename]
    push eax
    call 0x2a4c6        ; call the xbe!
xbename         db "d:\UDATA\4541000d\000000000000\linux.xbe",0
```

```
slot1:
        mov dl, 0x0F
        jmp short slot7

slot2:
        bswap edx
        jmp short slot1

slot3:
        xor al, cl
        jmp short slot21
```

```
00      80 ff d8 ff e0 00 10 4a 46 49 46 00 01 01 01 00 47 ˇÿˇ‡..JFIF.....G
00      90 00 47 00 00 ff fe 01 02 ad 53 fe 7d 78 85 0d 16 .G..ˇ˛..≠S˛}x...
0000:10a0 f8 05 c9 95 5d f5 4d 19 c0 84 3c a4 ab c0 d7 2f ¯.….]ıM.¿.<§´¿◊/
0000:10b0 2a f7 fa fa 6a 45 3c f2 b6 e7 c1 8e 83 10 4b f3 *˜˙˙jE<Ú∂Á¡...KÛ
0000:10c0 32 ca a3 19 e2 ef 5b cb f6 d8 88 5e a1 8c 6b c0 2 £.‚Ô[Àˆÿ.^°.k¿
0000:10d0 eb 8e 40 c3 d9 85 fb 41 e8 b2 fc ff ff c0 67 17 Î.@√Ÿ.˚AË≤˛ˇˇ¿g.
0000:10e0 bb 4d 4c 4b fa 69 fc be e9 ba 2c bd 42 41 40 aa ªMLK˙i˛æÈ∫‚ΩBA@™
```

Thank you Agent 007…

- Forum post on Xbox-Hacker BBS on March 29th 2003
- Since then Microsoft hasn't updated the game
- The buggy version is still in the shops



XboxHacker BBS ->General ->General Xbox Hacking & Modding

Pages: (6) < [1] 2 3 4 5 6 ... > (Go to first unread post)

Project B Solved!, Linux Project B solved!

« Next Oldest | Next Newest » Track this topic | Email this topic | Print this topic

habibi_xbox

Newbie

Group: Members
Posts: 1
Member No.: 8724
Joined: 29-March 03

Posted: Mar 29 2003, 12:20 PM

Subject : Project B Solved !

Ladies and Gentlemen,

I'm happy to present the first solution found for the Xbox Linux Project B:
Here is a way to run Xbox Linux on an unmodded, unopened Xbox !

Inlcuded is a uuencoded zip file containing all the necessary files. Here is
what you need:

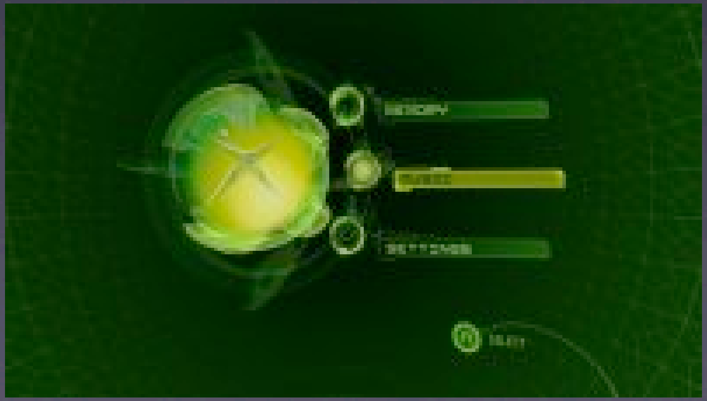
- - You need an unmodded XBOX (not sure it works with modded bios)
- - You need the game 007 Agent Under Fire (*NOT* NIGHTFIRE, those are two
different games!)
- - You need a way to transfer a save to a memory card (that is, xbox-save.com's
hardware, or usb<>xbox cable + usb stick + xbox-save software, or you can

Save Game Exploits

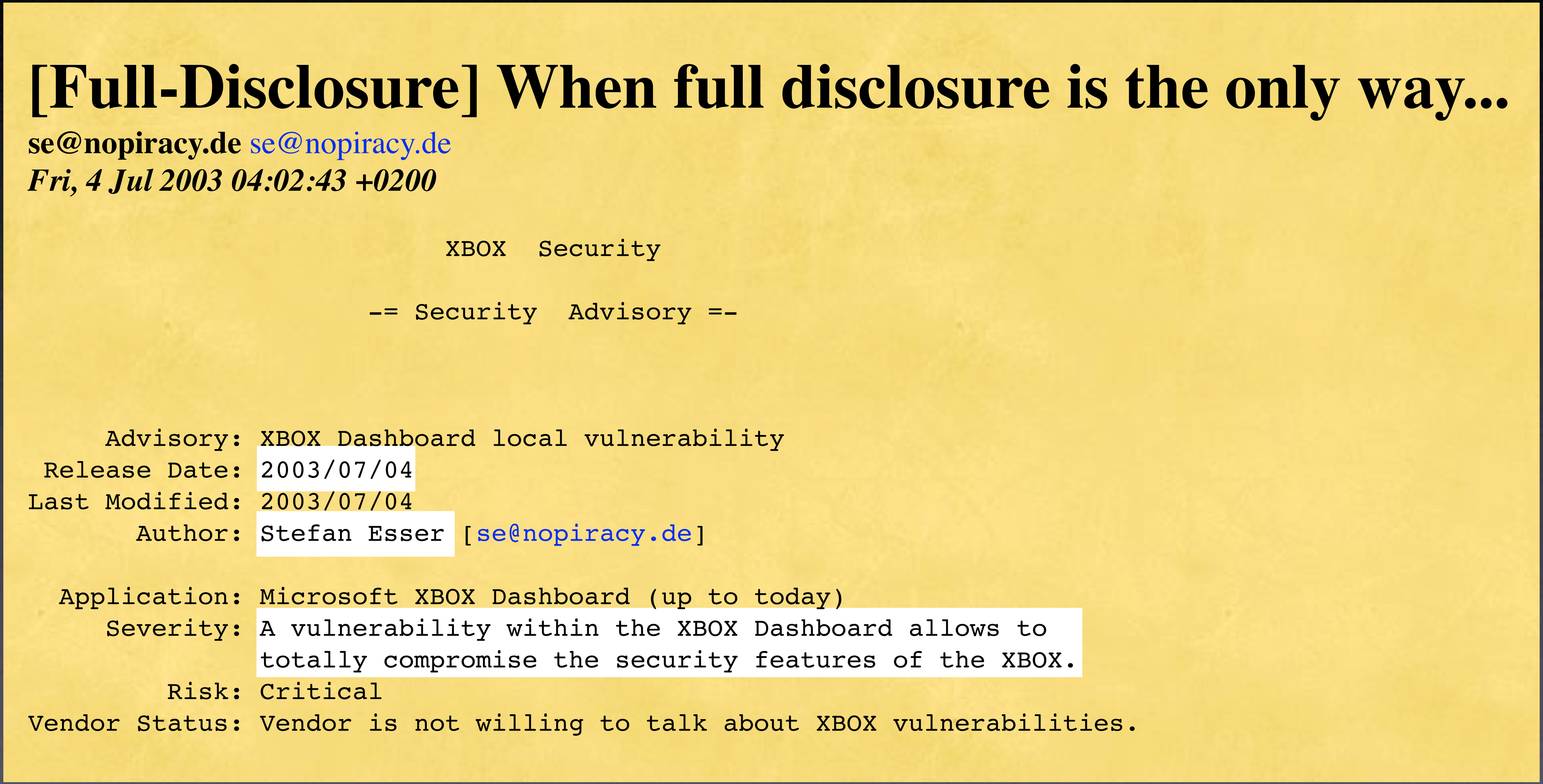
- Can be used for
  - Linux from memory card or hard disk
  - TSOP flashing
- Problem:
  - game needs to be started every time
  - DVD cannot be ejected

Dashboard

- the main program on hard disk
- gets run when there is no DVD
- drive can be ejected while Dashboard is running



Independence Day



[Full-Disclosure] When full disclosure is the only way...

se@nopiracy.de se@nopiracy.de

Fri, 4 Jul 2003 04:02:43 +0200

```
                           XBOX  Security

                      -= Security  Advisory =-



     Advisory: XBOX Dashboard local vulnerability
 Release Date: 2003/07/04
Last Modified: 2003/07/04
       Author: Stefan Esser [se@nopiracy.de]

  Application: Microsoft XBOX Dashboard (up to today)
     Severity: A vulnerability within the XBOX Dashboard allows to
               totally compromise the security features of the XBOX.
         Risk: Critical
Vendor Status: Vendor is not willing to talk about XBOX vulnerabilities.
```

Dashboard Font Exploit

Stefan Esser

Dashboard Used Files

Data

.XIP

Sounds

.WAV

Fonts

.XTF

Custom Soundtracks

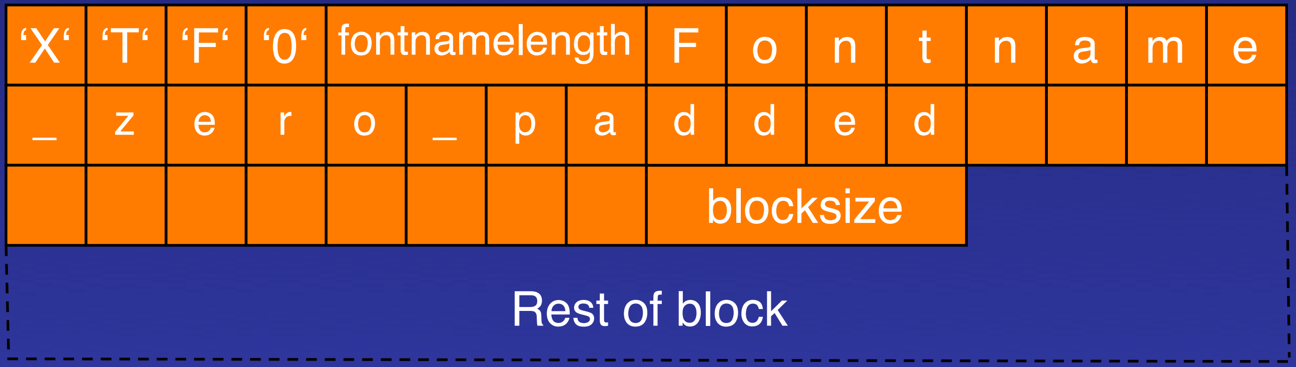
ST.DB

.WMA files

Savegames

.XBX (unicode .ini/bitmap format)

Font Loader Bug



```
...
ReadFile(handle, &dwBlockSize, 4, &dwNumRead, NULL);

pBlock = new BYTE [dwBlockSize];

*(DWORD*)pBlock = dwBlockSize;
ReadFile(handle, pBlock + 4, dwBlockSize – 4, &dwNumRead, NULL);
...
```

Overflow if dwBlockSize 0..3

Integer Underflow
(HEAP Overflow)

Bug Found – What Now?

Several questions

1. Is the XDK Heap Manager exploitable?

   → analysis of Xbox heap manager

2. How can we develop an exploit without a debugger?

   → trial and error with custom-made memory dumper

3. How can we get shellcode into the address space?

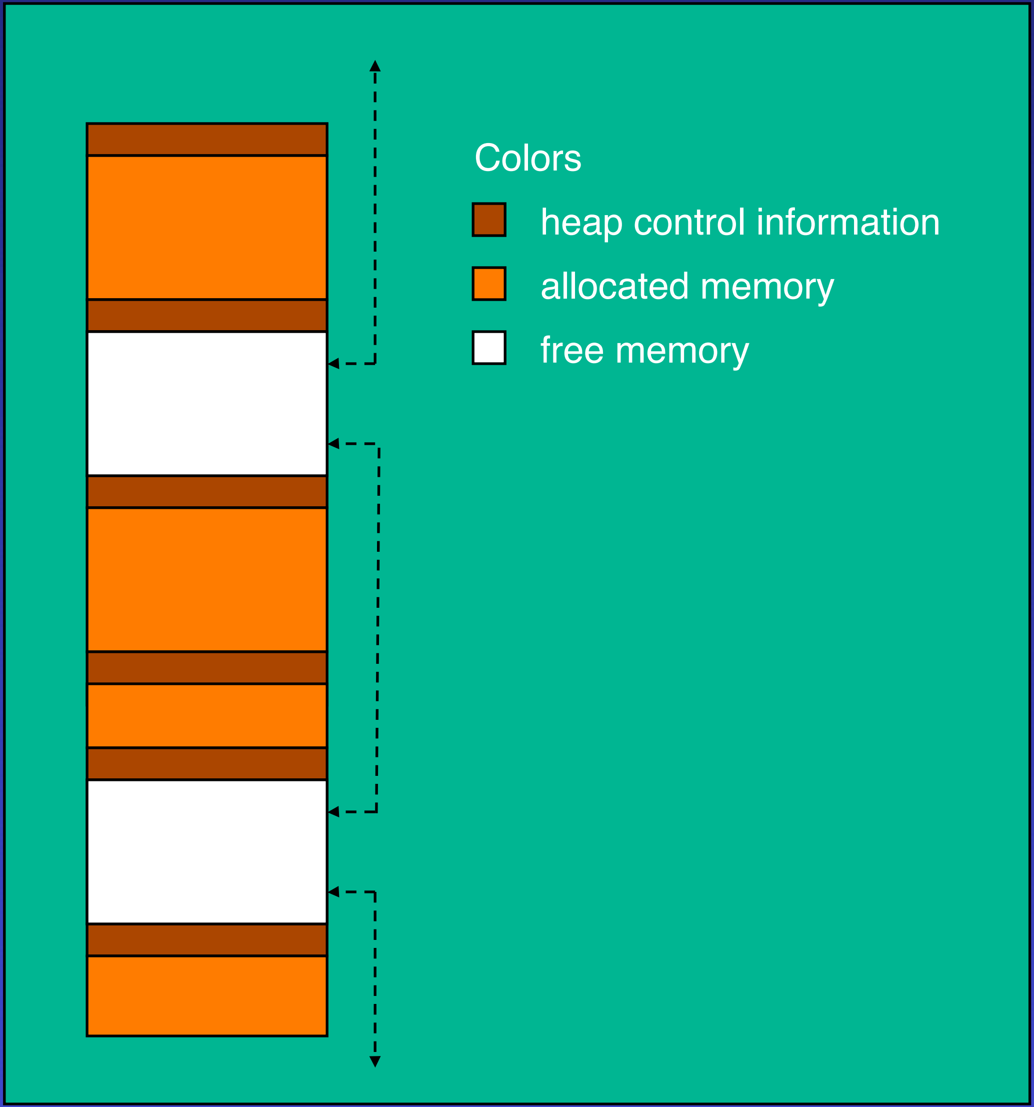
   → only through modified files

Heap on Xbox

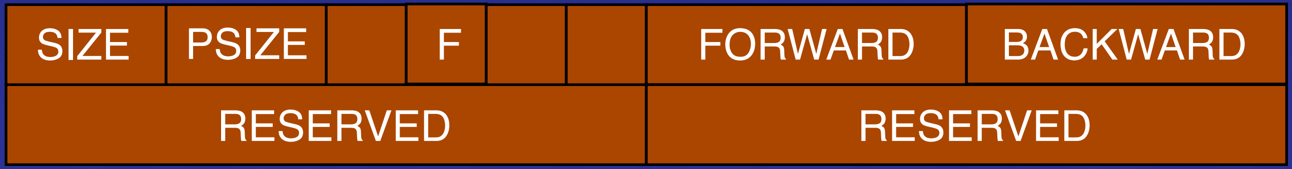
Heap control information stored together with data

Every block is aligned on 16 byte boundary

Free blocks in doubly-linked free lists



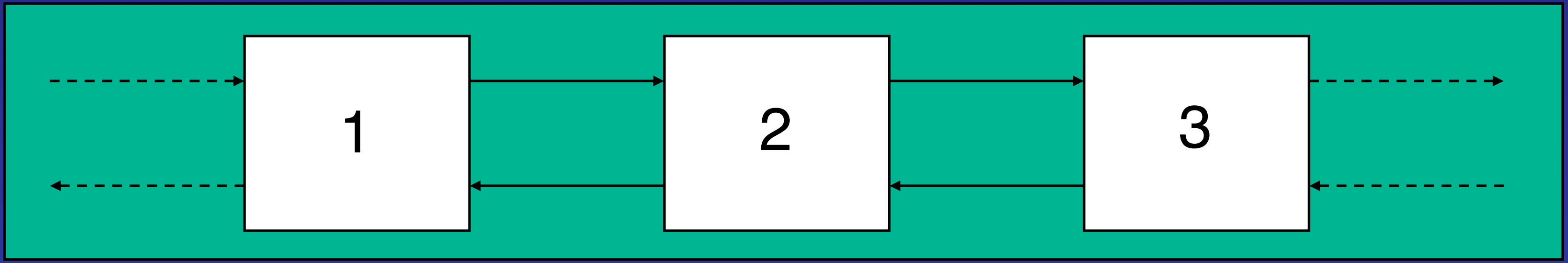
Xbox Heap: Free Block Header



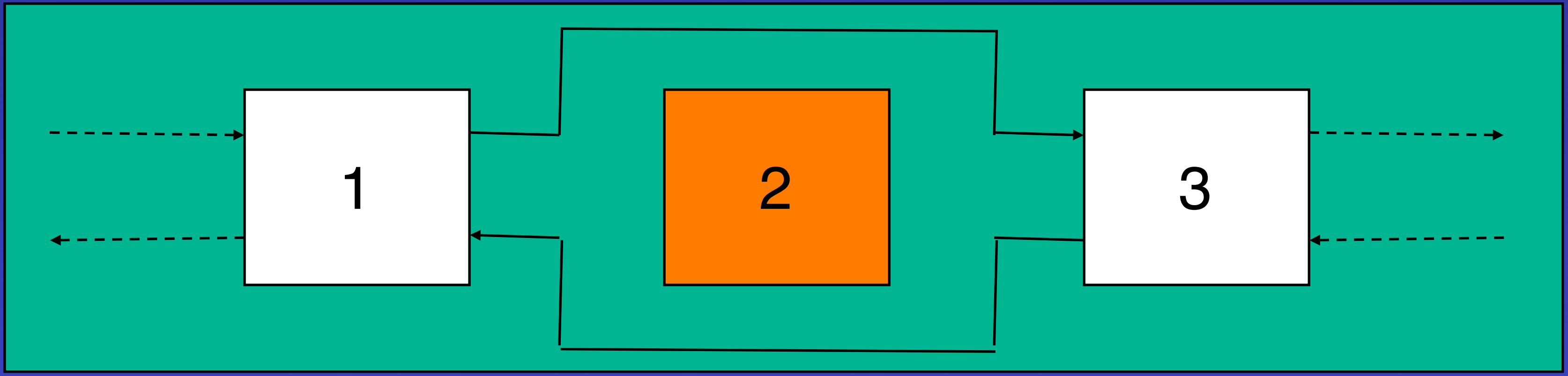
SIZE	Size of this block in 16 byte units	
PSIZE	Size of previous block in 16 byte units	
F	<u>Flags</u>	
	0x04	*Fill on free*
	0x10	*Last in list?*
FORWARD/BACKWARD		
	Next/previous free block of same size in linked list	

Xbox Heap: Unlink Operation

State before reallocation of block 2



State after reallocation of block 2



Pseudo-Unlink-Code:

```
B2→bwd→fwd = B2→fwd
B2→fwd→bwd = B2→bwd
```

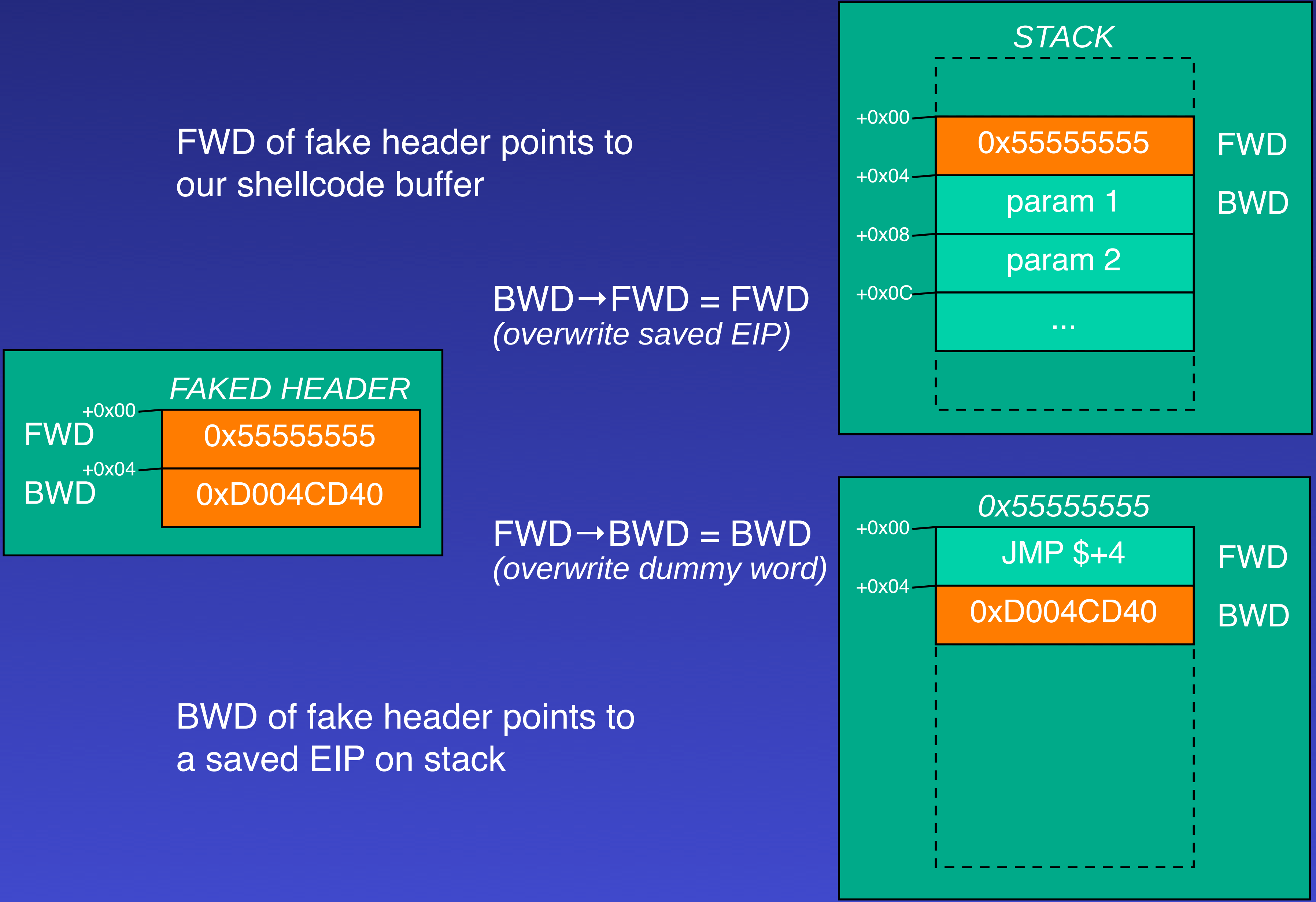
Xbox Heap: Unlink Example

FWD of fake header points to our shellcode buffer

BWD→FWD = FWD
(overwrite saved EIP)

FWD→BWD = BWD
(overwrite dummy word)

BWD of fake header points to a saved EIP on stack



Returning into Code – How?

- return address on stack?
- no, stack address is different for different threads

- imported function pointer?
- no, pages are write protected

- entry in x86 IDT?
- no, is unstable / depends on kernel

- Structured Exception Handler (SEH) chain on stack?
- yes, topmost SEH handler will get called

How to get code into address space?

- modify XBE (append/inject)?
- not possible because of digital signature

- inject into XIP data files?
- not possible, dashboard “knows” SHA1 hash

- inject into WAV audio files?
- were not found in memory dump

- store in another font file?
- yes, the only known solution (so far)

dayX shellcode

1. Find Kernel Base Address

   (Take entry from IDT and scan back until PE header found)

2. Lookup needed kernel exports *(only for LED flashing)*

3. Exchange RSA key *(Habibi modulus / exponent: 3)*

4. Play with LED color

5. Search for XDK xbe execution routine inside Dashboard

6. Call XDK routine to execute linux.xbe

Exploit Construction (overflow font)

1. Create malformed font to overwrite heap control information of following block

2. Use 0x55555555 and 0x66666666 as fwd/bwd pointers

3. Run and check memory/coredump

4. Repeat until heap control information is correctly overwritten, so that dashboard crashes with a write of 0x66666666 to 0x55555555

5. Simple because everything is aligned

Exploit Construction (shellcode font)

1. Create big font with large “NOP” sequence and shellcode

2. Dump memory and check where font is loaded

3. Repeat with different dashboards and different loading situations

4. Choose offset in the middle of Big Font to always hit “NOP” sequence

5. Put offset into “overflow font” (2nd offset is topmost SEH table address)

“Clock Loop” Problem

Unplugging box for several hours can result in Xbox rebooting over and over again

- No CMOS battery → losing clock settings
- Dashboard clock setup menu
- different code path
- lost thread race
- Crash/reboot

Paranoid Font Loader Fix

```
...
ReadFile_wrapper(handle, &dwBlockSize, 4, &dwNumRead, NULL);

if (dwBlockSize < MINIMUM_BLOCKSIZE) return (false);

pBlock = new BYTE [dwBlockSize];

*(DWORD*)pBlock = dwBlockSize;

ReadFile_wrapper(handle, pBlock + 4, dwBlockSize – 4, &dwNumRead, NULL);
...
```

Paranoid Font Loader Fix

```
...
ReadFile_wrapper(handle, &dwE

if (dwBlockSize < MINIMUM_BL

pBlock = new BYTE [dwBlockSiz

*(DWORD*)pBlock = dwBlockSize;

ReadFile_wrapper(handle, pBlock + 4, dwBlockSize – 4, &dwNumRead, NULL);
...
```

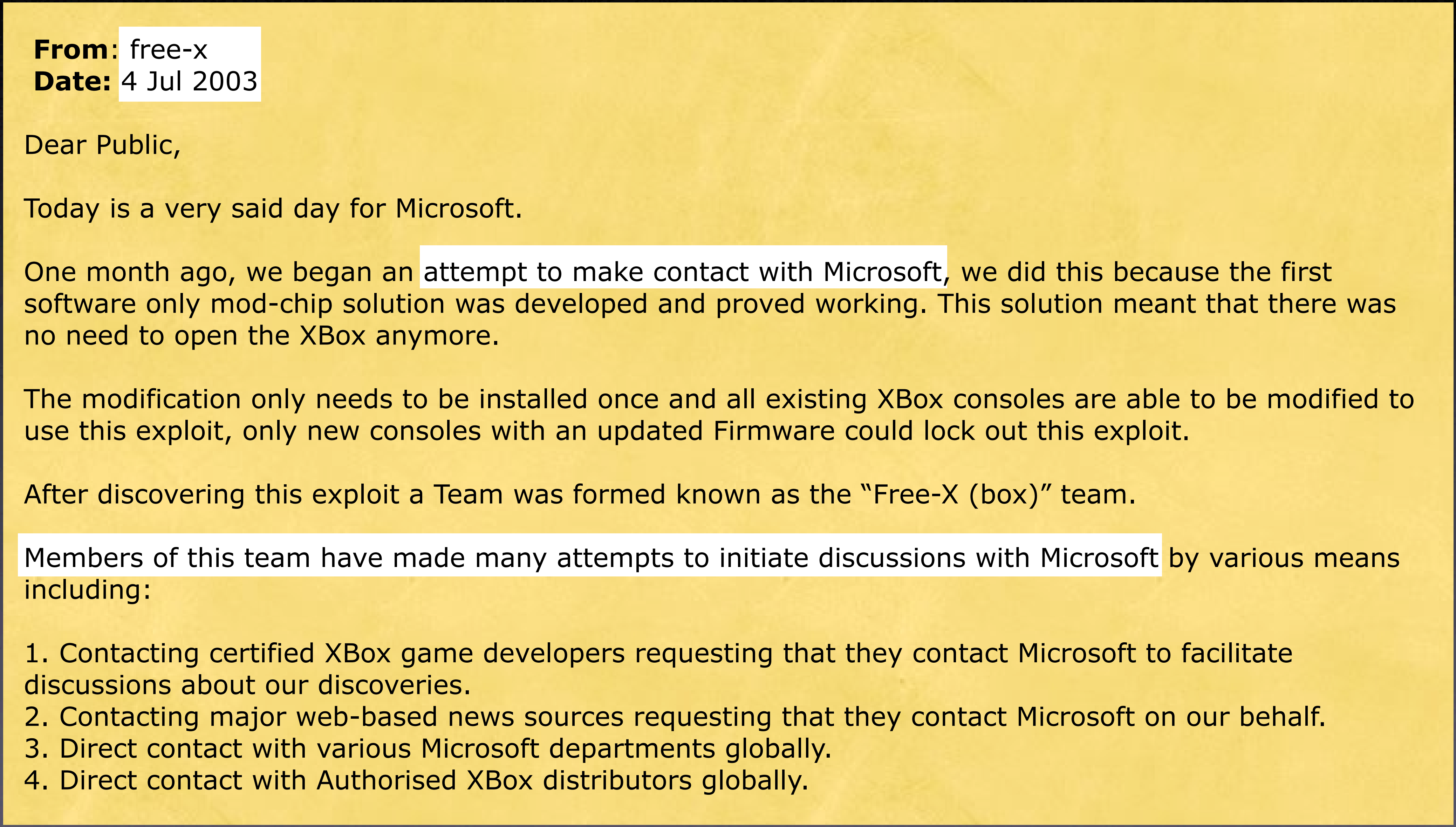
```
ReadFile_wrapper(p1, p2, length, p4, p5)
{
        if (length > 0x1400000) return (false);
        return ReadFile(p1, p2, length, p4, p5);
}
```

What now?

- the bug is severe
- it could be easily abused for piracy
- Microsoft had to be talked to

Okay, you guys... We're going to call Microsoft, and make our demands!
Linux on Xbox!
linux inside
Ahhhh!
What? What happened?
It was a girl! A girl answered!
They know our weakness.
You totally talked to her, though. That's pretty cool.
Yeah. I bet this is what sex feels like.
©2003 MIKE KRAHULIK AND JERRY HOLKINS
WWW.PENNY-ARCADE.COM

Talking to Microsoft



From: free-x
Date: 4 Jul 2003

Dear Public,

Today is a very said day for Microsoft.

One month ago, we began an attempt to make contact with Microsoft, we did this because the first software only mod-chip solution was developed and proved working. This solution meant that there was no need to open the XBox anymore.

The modification only needs to be installed once and all existing XBox consoles are able to be modified to use this exploit, only new consoles with an updated Firmware could lock out this exploit.

After discovering this exploit a Team was formed known as the “Free-X (box)” team.

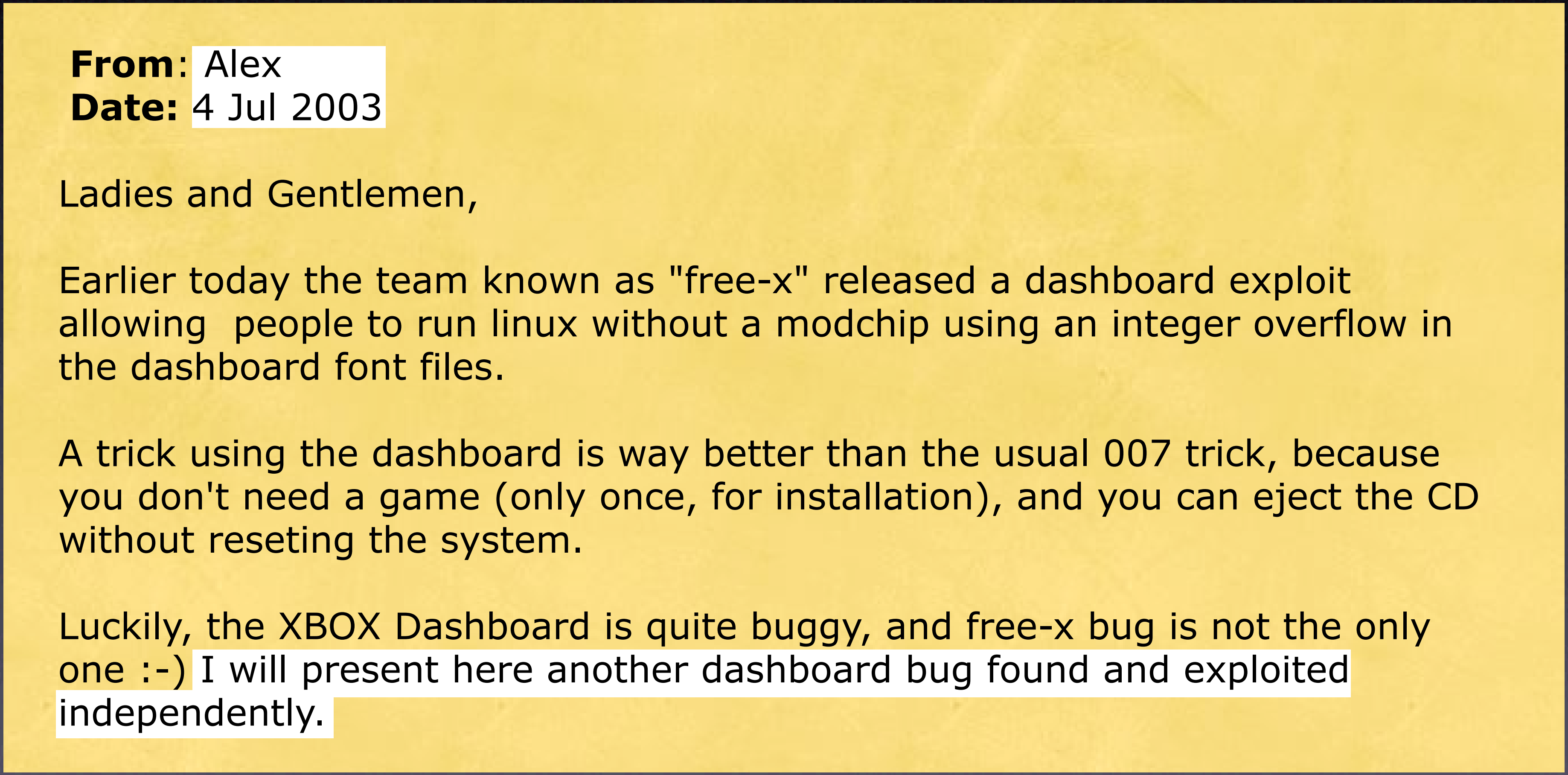
Members of this team have made many attempts to initiate discussions with Microsoft by various means including:

1. Contacting certified XBox game developers requesting that they contact Microsoft to facilitate discussions about our discoveries.
2. Contacting major web-based news sources requesting that they contact Microsoft on our behalf.
3. Direct contact with various Microsoft departments globally.
4. Direct contact with Authorised XBox distributors globally.

Speaking of the Dashboard...

- there is another thing...

Another Dashboard vulnerability



From: Alex
Date: 4 Jul 2003

Ladies and Gentlemen,

Earlier today the team known as "free-x" released a dashboard exploit allowing people to run linux without a modchip using an integer overflow in the dashboard font files.

A trick using the dashboard is way better than the usual 007 trick, because you don't need a game (only once, for installation), and you can eject the CD without reseting the system.

Luckily, the XBOX Dashboard is quite buggy, and free-x bug is not the only one :-) I will present here another dashboard bug found and exploited independently.

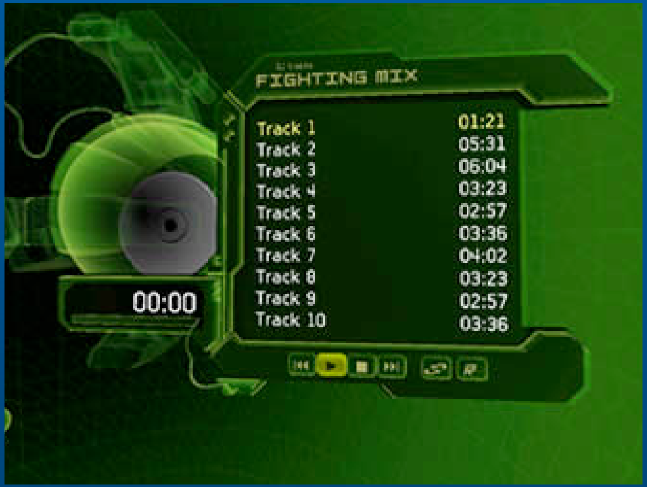
Dashboard Audio Exploit

- After releasing 007 hack, started working on a Dashboard hack
- Found one quickly
- Totally different bug than the one by Stefan Esser
- But same interest: no need to boot a game anymore, and CD can be ejected

Finding the Audio Exploit

- Most Dashboard files are signed, so can’t be modified.
- Idea : check « dynamic files » which can’t be signed
- Users can rip songs from CD and put them on the Xbox

- Songs indexed in a file named ST.DB



FIGHTING MIX

00:00

Track 1	01:21
Track 2	05:31
Track 3	06:04
Track 4	03:23
Track 5	02:57
Track 6	03:36
Track 7	04:02
Track 8	03:23
Track 9	02:57
Track 10	03:36

Inside ST.DB

- Contains lots of strings
- But no buffer overflow can be triggered on any string…
- Contains various integers, including the number of ripped tracks.
- No range check on this number!

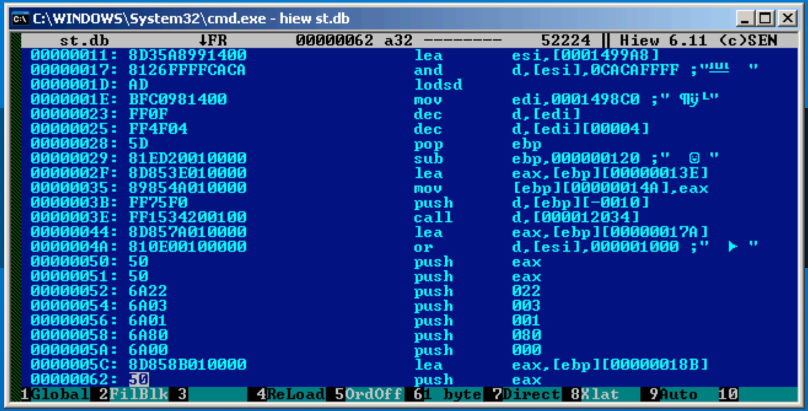
Let's exploit it!

- Inside the code, there is

  array[*number_of_tracks*] = *value*;

- Where both *number_of_tracks* and *value* come from the ST.DB file
- By putting a well-chosen *number_of_tracks* and *value*, we can write a 32 bit address at any place we want in RAM

Exploited…



```
   st.db          ↓FR          00000062  a32  ---------      52224 ‖ Hiew 6.11 (c)SEN
00000011: 8D35A8991400                    lea      esi,[0001499A8]
00000017: 8126FFFFCACA                    and      d,[esi],0CACAFFFF ;"╩╩  "
0000001D: AD                              lodsd
0000001E: BFC0981400                      mov      edi,0001498C0 ;" ¶ÿ└"
00000023: FF0F                            dec      d,[edi]
00000025: FF4F04                          dec      d,[edi][00004]
00000028: 5D                              pop      ebp
00000029: 81ED20010000                    sub      ebp,000000120 ;"  ☺ "
0000002F: 8D853E010000                    lea      eax,[ebp][00000013E]
00000035: 89854A010000                    mov      [ebp][00000014A],eax
0000003B: FF75F0                          push     d,[ebp][-0010]
0000003E: FF1534200100                    call     d,[000012034]
00000044: 8D857A010000                    lea      eax,[ebp][00000017A]
0000004A: 810E00100000                    or       d,[esi],000001000 ;"  ► "
00000050: 50                              push     eax
00000051: 50                              push     eax
00000052: 6A22                            push     022
00000054: 6A03                            push     003
00000056: 6A01                            push     001
00000058: 6A80                            push     080
0000005A: 6A00                            push     000
0000005C: 8D858B010000                    lea      eax,[ebp][00000018B]
00000062: 50                              push     eax
1Global 2FilBlk 3       4ReLoad 5OrdOff 61 byte 7Direct 8Xlat   9Auto  10
```

- For a given Dashboard version, ESP is always the same when this bug occurs
- Important so we can change the return address on the stack
- ~400 bytes of ST.DB are loaded to a constant address (for a given Dashboard version)
- We put our shellcode there

About Audio Exploit code

- The shellcode does the same as the 007 one
- But can’t be obfuscated in such a funny way because it must be 400 bytes maximum
- Code looks like it’s replacing the Xbox key with the 007 one
- But code uses self-modification, and in fact another key is used

Combining exploits

- a savegame exploit can run a small Linux
- the user can patch the hard disk for a Dashboard exploit
- Linux can be booted without the game, from hard disk

Exploits for Users

- automated installer needed
- ships as a memory card image
- combines MechAssault & Ernie/Bert
- converts the Dashboard into a boot menu

```
   _____                .__
  /     \   ____   ____ |  |__
 /  \ /  \_/ __ \_/ ___\|  |  \    ______
/    Y    \  ___/\  \___|   Y  \  /______/
\____|__  /\___  >\___  >___|  /
        \/     \/     \/     \/

.___                 __         .__  .__
|   | ____   _______/  |______  |  | |  |   ___________
|   |/    \ /  ___/\   __\__  \ |  | |  | _/ __ \_  __ \
|   |   |  \\___ \  |  |  / __ \|  |_|  |_\  ___/|  | \/
|___|___|  /____  > |__| (____  /____/____/\___  >__|
         \/     \/            \/               \/   v1.o
```

developed for the xbox-linux project by

Edgar "Gimli" Hueck

Franz "Solder" Lehner

Jeff "Kernel" Mears

Michael "Papa" Steil

Stefan "Exploit" Esser

and Kermit the Frog

MechInstaller

Dashboard Exploit Installer

Jeff Mears
asterisk@graces.dricas.com

Jeff Mears

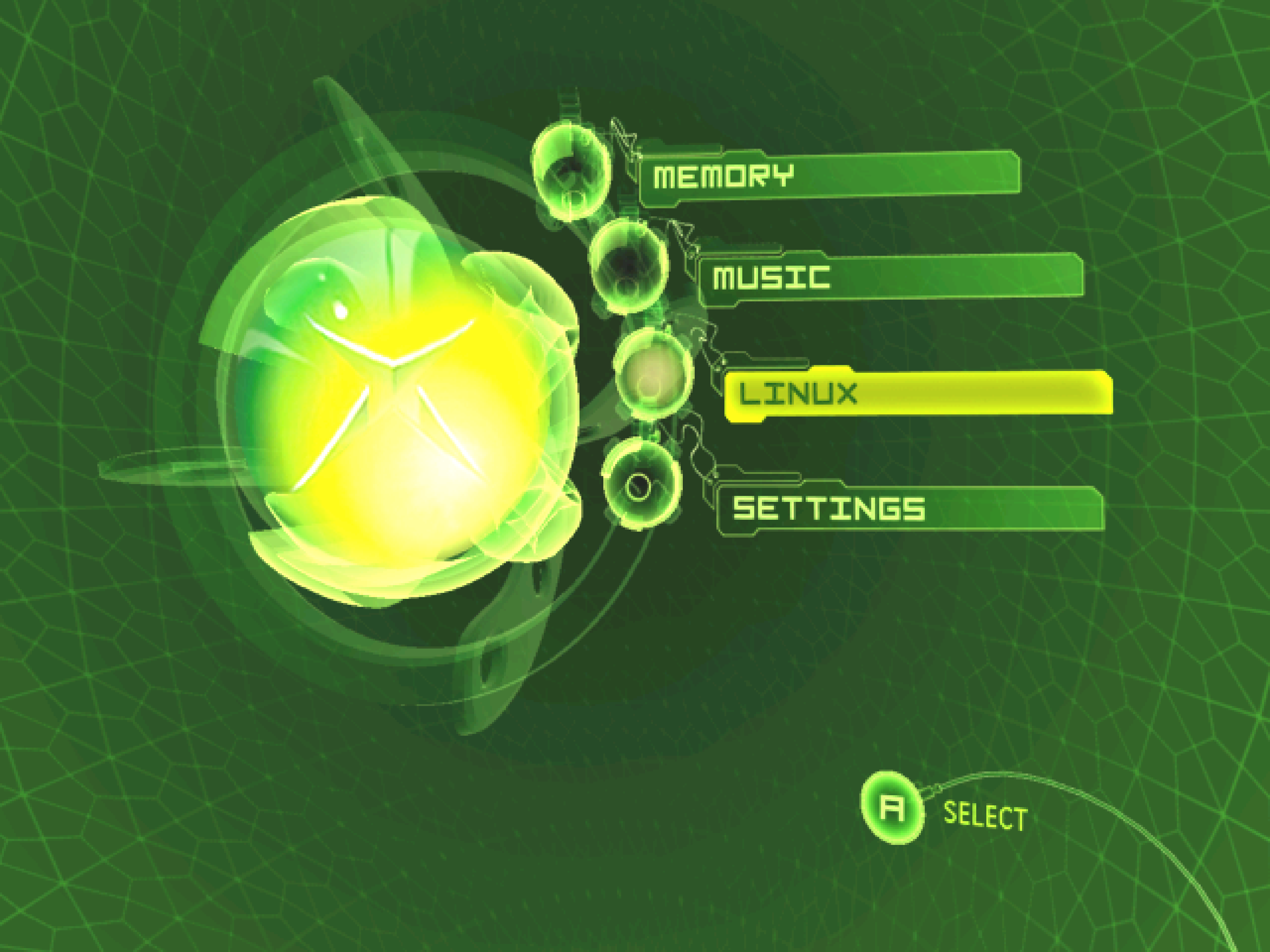
- Senior at University of California at Irvine
  - Graduates June 2004
- Experienced hacker of video games
  - Making cheat codes (Action Replay)
  - Translating old games to English
  - Emulation
- Xbox Linux contribution: MechInstaller, boot loader algorithm, Xbox kernel information

What is MechInstaller? (1)

- MechInstaller is a saved game exploit for the game MechAssault that installs Stefan's Dashboard exploit
- Automated installation process
- Minimal technical expertise needed
- All you need is a way to get save files onto the Xbox
  - Memory card adaptors for PC
  - A friend's modified Xbox

What is MechInstaller? (2)

- Regular functionality (except Xbox Live) of the Xbox as a game system is preserved
- Dashboard remains, but the “Xbox Live” menu option is replaced with “Linux”
- MechInstaller installs a Mini-Linux like 007
- With MechInstaller installed, Linux installation CDs (Xebian) boot directly so a permanent and full Linux can be installed


MEMORY
MUSIC
LINUX
SETTINGS
A
SELECT

Why?

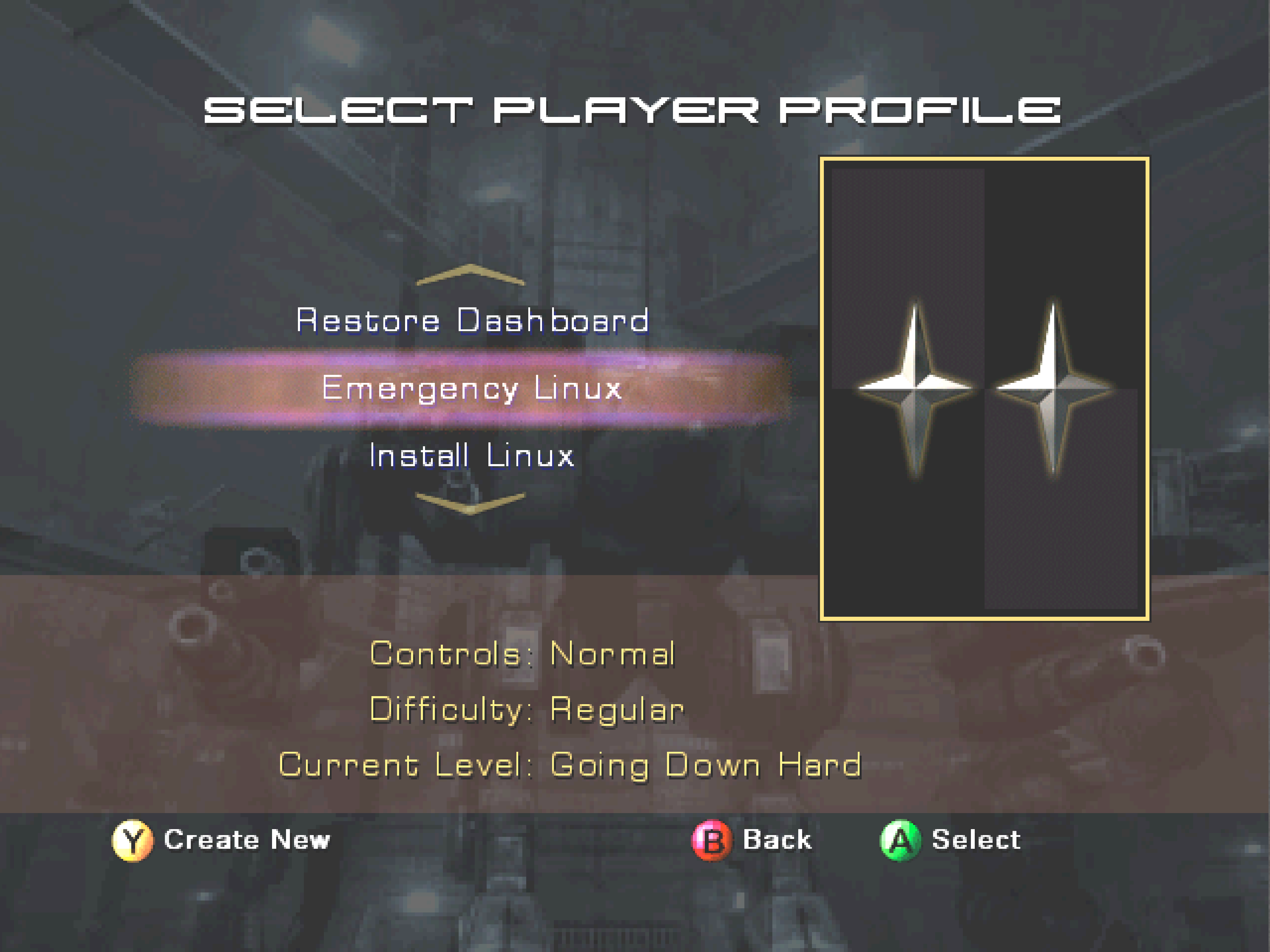
- Stefan's exploit is great, but difficult to use
  - Must boot 007 and do manual commands
  - Very tedious; cannot be done *en masse*
  - Difficult to recover from mistakes
- Multiple Dashboard versions
  - Stefan's exploit is most stable with one certain version
- Stefan's exploit completely replaces Dashboard, which many users don't want

MechInstaller as a Solution

- Difficulty: MechInstaller is fully automated and runs with the press of a button
- Recoverability: MechInstaller provides a “Restore Dashboard” option
  - Also provides emergency Mini-Linux like 007
- Compatibility: MechInstaller replaces the user’s existing Dashboard with known version regardless of previous version

Usability

- User loads memory card with hacked save
  - Same as the 007 exploit
- When booting MechAssault, the user sees a 3 option menu
  - Install Linux: Installs the Stefan exploit
  - Restore Dashboard: Install unmodified Dashboard
  - Emergency Linux: Runs a 007-like mini-Linux
- All options are fully automated



SELECT PLAYER PROFILE

Restore Dashboard

Emergency Linux

Install Linux

Controls: Normal

Difficulty: Regular

Current Level: Going Down Hard

Installing a Dashboard

- Installing a particular Dashboard version legally - but how?
- Solution: We have Microsoft do it for us.
- Xbox Live games have a copy of the Dashboard to install the Live Dashboard
- We use this installer (dashupdate.xbe) from the game DVD to install it

Why MechAssault?

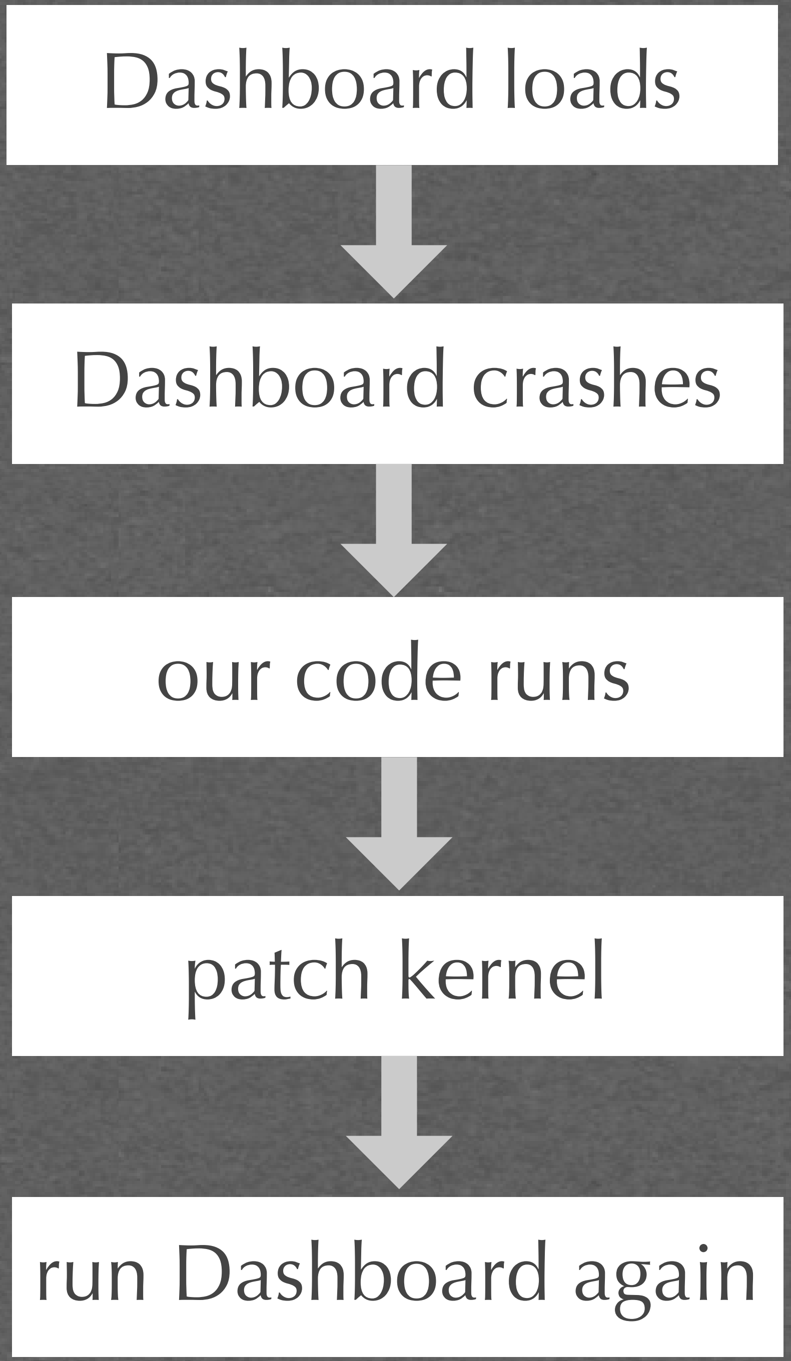
- Xbox Live game (has dashupdate.xbe)
- Easy and obvious saved game exploit
  - Classic sprintf() overflow on movie filenames
- Very popular and easy to find
- Multi-Region (both America and Europe)
- Exploit triggers *after* selecting save file
  - Allows the creation of a little menu by having multiple saved game exploits

How – The Installer

- Get control of Xbox through exploit
- Hook the XBE loader in the Xbox kernel
  - Must hook kernel to hook a new XBE file
- "Quick Reboot" and load dashupdate.xbe
- Kernel calls us right before XBE executes
  - Hook end-of-installation in dashupdate
  - Modify on-screen messages to user
- When we get control again, install hack
  - Copy mini-Linux to Dashboard partition also
  - Reboot system and eject the MechAssault disk

How – The Dashboard Exploit

- Get control at startup with Stefan's exploit
- However, the exploit badly corrupts heap
  - Cannot continue loading Dashboard
  - When exploit triggers, "Quick Reboot" into the Dashboard to reload it, except with a hook installed (and normal font files used)
- Modify "Xbox Live" text to say "Linux"
- Hook kernel's "Run XBE" routine
  - Called when Dashboard wants to run a disk
  - Also called when Dashboard tries to run Live (when the "Xbox Live" option, now "Linux", is chosen)


Dashboard loads
Dashboard crashes
our code runs
patch kernel
run Dashboard again

How – XBE Loading

- Our code is called when a disk is inserted
  - Any non-game disk will also trigger this
- 3 cases
  - DVD movie or Xbox game inserted: Load normally
  - Linux menu option chosen: Load Linux
  - Linux CD/DVD inserted: Boot CD
- How to load Linux?
  - Detect Linux CDs from a special marking in the XBE header (not secure, but if the XBE were not made by us, the signature check would fail later when loading)
  - Replace the public key like 007

Security

- Modifying MechInstaller to install a pirate game loader is quite tempting to pirates
  - Must prevent modifications like this
- Like 007, we take a two-pronged approach
  - Obfuscate the exploit engine (done by Stefan)
  - Sign the XBEs we load with our own key
- Security eventually made useless by hacks
  - Phoenix BIOS Loader provided a much better pirate solution than MechInstaller, making it obsolete
  - MechInstaller eventually cracked to load anything

Key Replacement (1)

- Completely replacing the RSA key used to sign XBEs with our own would be nice
  - Only need to change key to load our XBEs – don't need to delete kernel's security checks
  - However, certain parts of the XBE header are encrypted directly from the key
    - 16 bytes (out of 256) used as a simple XOR key
    - Changing this XOR is okay on unmodified systems. However, because mod chips disable this security, this won't work properly! Linux CDs must be bootable by both MechInstaller and mod chip users.

Key Replacement (2)

- First idea: Leave RSA modulus alone but set the “public exponent” to 1
  - RSA equivalent of a “null” key: encrypted data equals decrypted data. “Signing” is a no-op.
  - Works, but very easy to repeat
- 007: Change the modulus to a weak one
  - Modify parts that don’t affect the header XOR
  - Make the modulus into either a prime number or an easily factored number (the latter was used in 007)
  - 007’s key factored before exploit was cracked

Key Replacement (3)

- MechInstaller: Create a “strong” key with the same XOR as the original
  - Take advantage of the fact that the XOR key comes from the *middle* part of the modulus
  - The key made this way is not as secure as normal RSA-2048
    - Much easier to reverse engineer obfuscation than to factor this key

Do as they do

- we do obfuscation and RSA key replacement
- this technology can possibly be abused
- the source of MechInstaller needs to be kept secret!

Microsoft's reaction

- 25 Jun 2003: MechAssault exploit
- 4 Jul 2003: Dashboard exploits
- 11 Aug 2003: MechInstaller released
- 12 Sep 2003: Dashboard update (Xbox Live)
- Nov 2003: fixed versions of MechAssault

Dashboard Update

- Microsoft updates the Dashboard remotely
- without asking
- even if you’re not an Xbox Live user
- can’t be undone without the old image
- so don’t buy Xbox Live
- and don’t enter network settings in the Dashboard

Can they just...?

- they say running Linux on the Xbox is illegal
- they say the forced update is legal
- they say you lose the right to use the old version

Can I just...?

If it’s legal for them to force you to upgrade, wouldn’t it be legal for any user to downgrade?

Hacking Status

- most new Xboxes still contain the vulnerable dashboard
- else it is easy to downgrade
- there are a million vulnerable MechAssault DVDs out there
- and did we mention...

there is more...